

Amendments to the Claims

1. (Currently Amended) A method for preventing an unread activity from being bounced-back to an originating server during a replication operation, comprising:

storing an identification of an originating server of a replicated unread activity in an unread log of a receiving server; and

during a subsequent replication process initiated by the receiving server, preventing replication of the unread activity back to the originating server;

during the subsequent replication process, replicating the unread activity to at least one other server not identified as the originating server;

wherein storing an identification further comprises updating the unread log to include an unread entry corresponding to the replicated unread activity, and

storing the identification of the originating server with the unread entry; and

wherein preventing the replication of the unread activity back to the originating server further comprises examining the unread log to determine if any unread entries stored therein correspond to an unread activity received from the originating server and, during the subsequent replication process, not replicating any unread activity identified as being received from the originating server back to the originating server.

Claims 2-4 (Cancelled).

5. (Original) The method of claim 1, wherein the originating server has a name, and wherein the identification is a hash of the name of the originating server.
6. (Original) The method of claim 5, wherein during the subsequent replication process, if another server has the same hash as the originating server, the receiving server replicates the unread activity to the other server and back to the originating server.
7. (Original) The method of claim 6, wherein the originating server discards any duplicate replicated unread activities.
8. (Currently Amended) A bounce-back prevention system, comprising:
 - a receiving server for receiving an unread activity replicated by an originating server, the receiving server including an unread log for storing an identification of the originating server; and
 - a system for preventing replication of the unread activity back to the originating server during a subsequent replication process initiated by the receiving server;

wherein the receiving server further comprises a replication system, and
wherein the replication system of the receiving server replicates the unread
activity to at least one other server not identified as the originating server during
the subsequent replication process;

wherein the receiving server further comprises a system for updating the unread log to include an unread entry corresponding to the replicated unread activity, and for storing the identification of the originating server with the unread entry; and

wherein the system for preventing the replication of the unread activity back to the originating server further comprises a system for examining the unread log to determine if any unread entries stored therein correspond to an unread activity received from the originating server, and a system for preventing any unread activities, identified by the examining system as being received from the originating server, from being replicated back to the originating server, during the subsequent replication process.

Claims 9-11 (Cancelled).

12. (Original) The system of claim 8, wherein the originating server has a name, and wherein the identification is a hash of the name of the originating server.

13. (Original) The system of claim 12, wherein the receiving system includes a replication system, and wherein during the subsequent replication process, if another server has the same hash as the originating server, the replication system of the receiving server replicates the unread activity to the other server and back to the originating server.

14. (Original) The system of claim 13, wherein the originating server discards any duplicate replicated unread activities.

15. (Currently Amended) A program product stored on a recordable medium for preventing an unread activity from being bounced-back to an originating server during a replication operation, which when executed on a computer system comprises:

program code for storing an identification of an originating server of a replicated unread activity in an unread log of a receiving server; and

program code for preventing replication of the unread activity back to the originating server, during a subsequent replication process initiated by the receiving server;

program code for replicating the unread activity to at least one other server not identified as the originating server, during the subsequent replication process;

wherein the program code for storing an identification further comprises program code for updating the unread log to include an unread entry corresponding to the replicated unread activity, and program code for storing the identification of the originating server with the unread entry; and

wherein the program code for preventing the replication of the unread activity back to the originating server further comprises program code for examining the unread log to determine if any unread entries stored therein

correspond to an unread activity received from the originating server, and
program code for preventing replication of any unread activity the examining
program code has identified as being received from the originating server back to
the originating server, during the subsequent replication process.

Claims 16-18 (Cancelled).

19. (Original) The program product of claim 15, wherein the originating server has a name, and wherein the identification is a hash of the name of the originating server.

20. (Original) The program product of claim 19, wherein the receiving server further includes program code for replicating the unread activity to the other server and back to the originating server during the subsequent replication process, if another server has the same hash as the originating server,

21. (Original) The program product of claim 20, wherein the originating server includes program code for discarding any duplicate replicated unread activities.

22. (Cancelled).